****

**AI ASSISTED CODING**

**LAB-7: *Error Debugging with AI – Systematic Approaches to Finding and Fixing Bugs***

**Roll no:** 2503A51L34

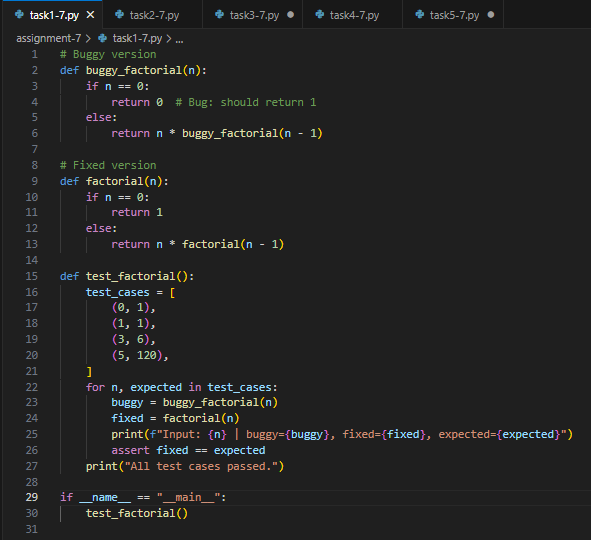
**Name:** Uzma Yasmeen

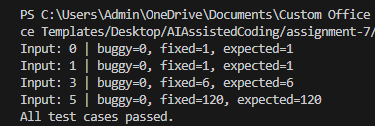
**Batch:** 25BTCAICSB20

**Task-1 Description:** Introduce a buggy Python function that calculates the factorial of a number using recursion. Use Copilot or Cursor AI to detect and fix the logical or syntax errors.

**Prompt:** Introduce a buggy Python function that calculates the factorial of a number using recursion. Then detect and fix the logical or syntax errors.

**Code Generated:**

****

**Output:**

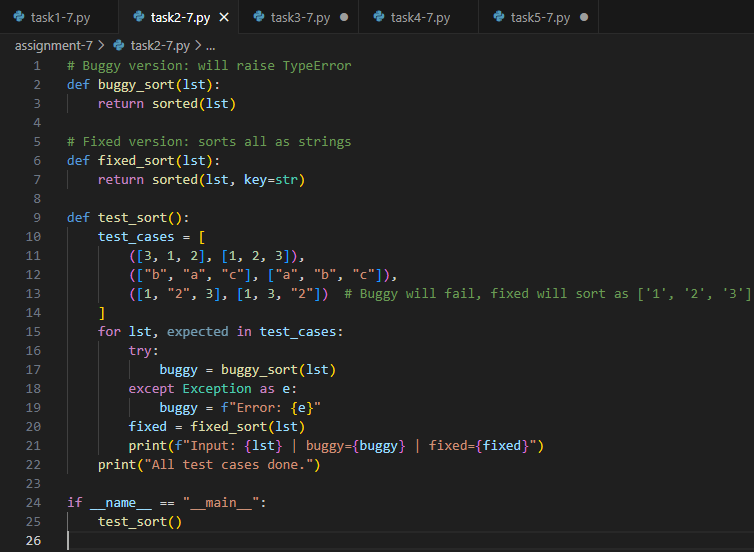
**Observation:**

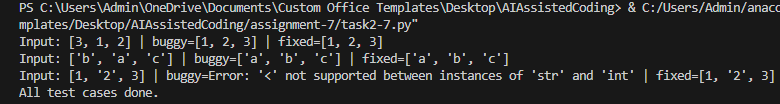
In this task, a recursive factorial function was introduced with errors. AI helped identify and fix the logical issues. This showed how debugging tools can simplify correcting syntax and logic mistakes.

**Task-2 Description:** Provide a list sorting function that fails due to a type error (e.g., sorting list with mixed integers and strings). Prompt AI to detect the issue and fix the code for consistent sorting.

**Prompt:** Provide a list sorting function that fails due to a type error (e.g., sorting list with mixed integers and strings). Detect the issue and fix the code for consistent sorting.

**Code Generated:**

****

**Output:**

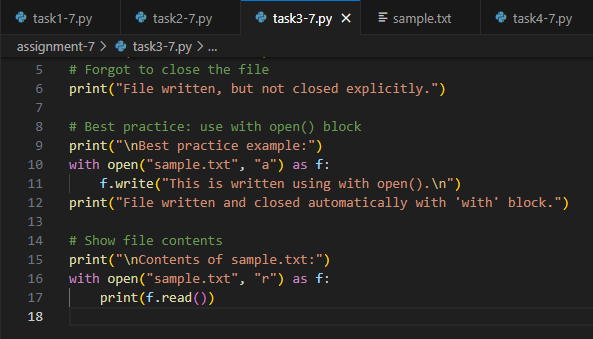
**Observation:**

The sorting function failed due to mixed data types. AI detected the type error and corrected the code to allow consistent sorting. It highlighted the importance of handling data types properly in Python.

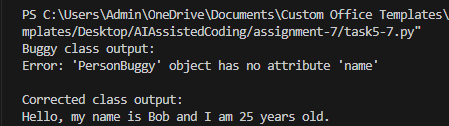
**Task-3 Description:** Write a Python snippet for file handling that opens a file but forgets to close it. Ask Copilot or Cursor AI to improve it using the best practice (e.g., with open() block).

**Prompt:** Write a Python snippet for file handling that opens a file but forgets to close it. Improve it using the best practice (e.g., with open() block).

**Code Generated:**

****

**Output:**

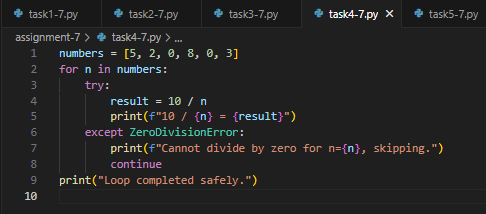
****

**Observation:**

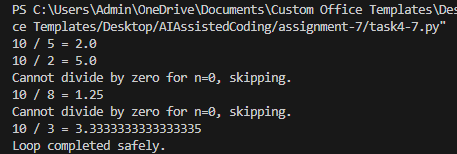
A file handling code was written without closing the file. AI suggested using the with open() block, which automatically manages file closing. This improved the code and ensured best practices in file handling.

**Task-4 Description:** Provide a piece of code with a ZeroDivisionError inside a loop. Ask AI to add error handling using try-except and continue execution safely.

**Prompt:** Generate a code that Provide a piece of code with a ZeroDivisionError inside a loop. Add error handling using try-except and continue execution safely.

**Code Generated:**

**Output:**

****

**Observation:**

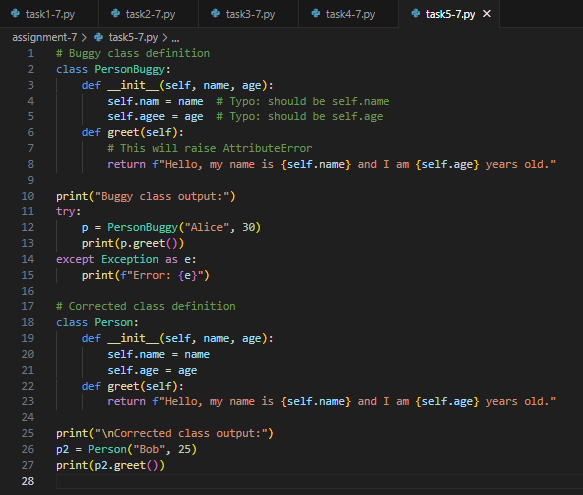
A loop contained a ZeroDivisionError, which caused program failure. AI added try-except handling so the loop could continue safely. This demonstrated how exception handling prevents program crashes.

**Task-5 Description:** Include a buggy class definition with incorrect \_\_init\_\_ parameters or attribute references. Ask AI to analyze and correct the constructor and attribute usage.

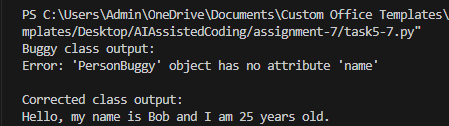
**Prompt:** Write a python script which Includes a buggy class definition with

incorrect init parameters or attribute references. Analyze and correct the constructor and attribute usage.

**Code Generated:**

****

**Output:**

****

**Observation:**

A buggy class with incorrect constructor parameters was introduced. AI analyzed the class and fixed the errors in \_\_init\_\_ and attribute usage. This improved understanding of class structure and object initialization.